Overview of the LinearAlgebra Package

Basic Functionality

Description

- The **LinearAlgebra** package offers routines to construct and manipulate Matrices and Vectors, compute standard operations, query results and solve linear algebra problems.

- For a complete list of the routines in the **LinearAlgebra** package, see the Details of the Linear Algebra Package help page.

Output

- Matrix, Vector, or an expression sequence of the two. Matrices 10 x 10 or smaller and vectors 10 x 1 and smaller display the corresponding Matrix or Vector in the Maple worksheet. Matrices and vectors larger then this display a placeholder as output. To see the entries or structured views of the Matrix or Vector, double-click the placeholder. For more details, see the browse Matrix help page.

Interfaces to the LinearAlgebra Package

Commands

- Each command in the **LinearAlgebra** package is accessed by using either the long form or the short form of the command name in the command calling sequence. For more information, see the Using Packages help page.

Long form

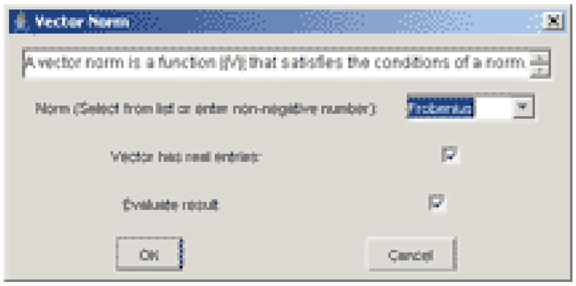
```
> LinearAlgebra[RandomMatrix](2);
```

Short form

```
> with(LinearAlgebra):
  RandomMatrix(2);
```

Maplets

- Some routines in the **LinearAlgebra** package come with Maplet interfaces. To see the available interfaces, see the Maplets[Examples][LinearAlgebra] help page.



Vector Norm

A vector norm is a function ||V|| that satisfies the conditions of a norm.

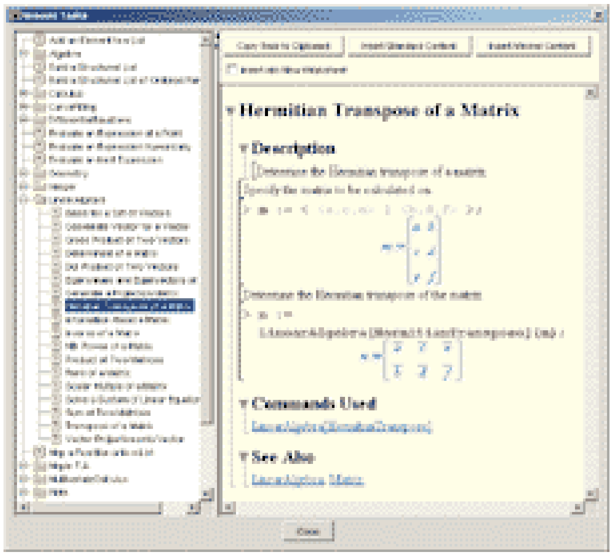
Norm (Select from list or enter non-negative number): Frobenius

Vector has real entries: ☑

Evaluate result ☑

Task

- Some routines in the **LinearAlgebra** package come with a task template to step you through the process of solving a linear algebra problem. For more information, see the Using Tasks help page.



Hermitian Transpose of a Matrix

Description

Determine the Hermitian transpose of a matrix.

[illegible]

Determine the Hermitian transpose of the matrix.

Commands Used

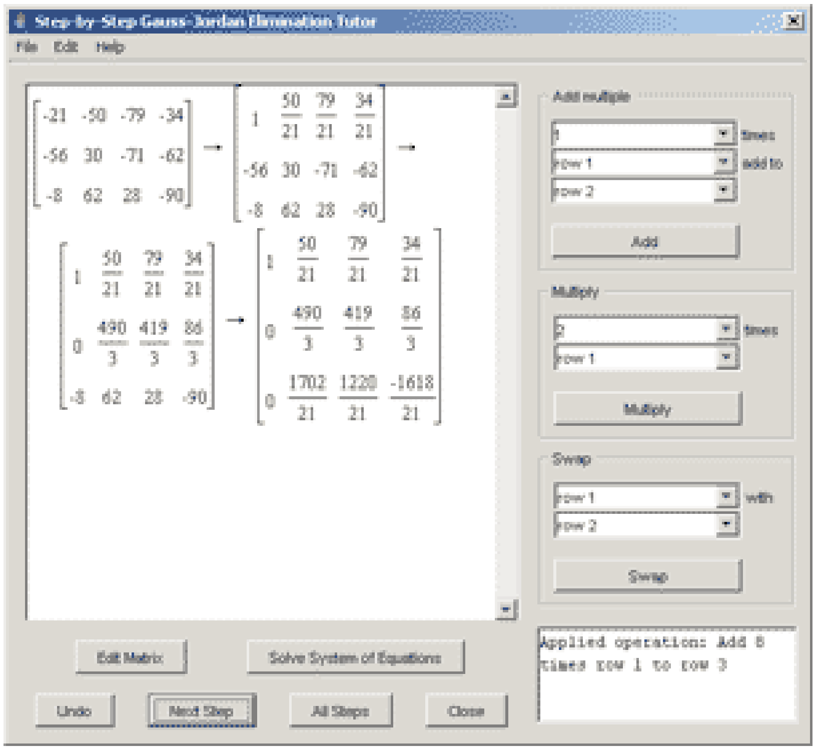
LinearAlgebra[HermitianTranspose]

See Also

LinearAlgebra, Matrix

Student[LinearAlgebra] Package

- For students learning the concepts presented in an introductory linear algebra course, see the Student[LinearAlgebra] help page.



Step-by-Step Gauss-Jordan Elimination Tutor

$$\begin{bmatrix} -21 & -50 & -79 & -34 \\ -56 & 30 & -71 & -62 \\ -8 & 62 & 28 & -90 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & \frac{50}{21} & \frac{79}{21} & \frac{34}{21} \\ -56 & 30 & -71 & -62 \\ -8 & 62 & 28 & -90 \end{bmatrix} \rightarrow$$

$$\begin{bmatrix} 1 & \frac{50}{21} & \frac{79}{21} & \frac{34}{21} \\ 0 & \frac{490}{3} & \frac{419}{3} & \frac{86}{3} \\ -8 & 62 & 28 & -90 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & \frac{50}{21} & \frac{79}{21} & \frac{34}{21} \\ 0 & \frac{490}{3} & \frac{419}{3} & \frac{86}{3} \\ 0 & \frac{1702}{21} & \frac{1220}{21} & \frac{-1618}{21} \end{bmatrix}$$

Add multiple: 1 times row 1 add to row 2

Multiply: 2 times row 1

Swap: row 1 with row 2

```
Applied operation: Add 8
times row 1 to row 3
```

Essential LinearAlgebra Package Commands

Basis	return a basis for a vector space
CharacteristicPolynomial	construct the characteristic polynomial of a Matrix
CrossProduct	compute the cross product of two Vectors
DeleteRow	delete rows of a Matrix
Determinant	compute the determinant of a Matrix
Dimension	determine the dimension of a Matrix or a Vector
DotProduct	compute the dot product of two Vectors
Eigenvalues	compute the eigenvalues of a Matrix
Eigenvectors	compute the Eigenvectors of a Matrix
GaussianElimination	perform Gaussian elimination on a Matrix
LeastSquares	compute the least-squares to equations
LinearSolve	solve the linear equations A . x = b
Map	map a procedure onto an expression
MatrixInverse	compute the inverse of a square Matrix
MatrixScalarMultiply	compute the product of a Matrix and a scalar
NullSpace	compute a basis for the nullspace of a Matrix
RandomMatrix	construct a random Matrix
ReducedRowEchelonForm	perform Gauss-Jordan elimination on a Matrix
SubMatrix	construct a submatrix of a Matrix
Transpose	compute the transpose of a Matrix

Examples

```
> with(LinearAlgebra):
```

Construct a 5 x 5 Matrix.

```
> M:=RandomMatrix(5);
```

$$M := \begin{bmatrix} -81 & -98 & -76 & -4 & 29 \\ -38 & -77 & -72 & 27 & 44 \\ -18 & 57 & -2 & 8 & 92 \\ 87 & 27 & -32 & 69 & -31 \\ 33 & -93 & -74 & 99 & 67 \end{bmatrix} \qquad \textbf{(4.1)}$$

Construct a submatrix of the Matrix **M**, where the first list in the calling sequence selects corresponding row entries and the second list selects column entries.

```
> SubMatrix(M, [2..5], [2..3, 1]);
```

$$\begin{bmatrix} -77 & -72 & -38 \\ 57 & -2 & -18 \\ 27 & -32 & 87 \\ -93 & -74 & 33 \end{bmatrix} \qquad \textbf{(4.2)}$$

Construct the Sylvester Matrix of two polynomials.

```
> SylvesterMatrix(x^2+3*x, 2*x);
```

$$\begin{bmatrix} 1 & 3 & 0 \\ 2 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix} \qquad \textbf{(4.3)}$$

Compute the Eigenvectors of a Matrix.

```
> Eigenvectors(Matrix([[4, -1, 6], [2, 1, 6], [2, -1, 8]]));
```

$$\left[\begin{array}{c} 9 \\ 2 \\ 2 \end{array}\right], \left[\begin{array}{ccc} 1 & -3 & \frac{1}{2} \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{array}\right] \tag{4.4}$$

Test if the Matrix **M** is orthogonal.

```
> M:=Matrix([[sqrt(10)*3/10, -sqrt(10)/10], [sqrt(10)/10, sqrt
  (10)*3/10]]);
```

$$M := \left[\begin{array}{cc} \frac{3}{10}\sqrt{10} & -\frac{1}{10}\sqrt{10} \\ \frac{1}{10}\sqrt{10} & \frac{3}{10}\sqrt{10} \end{array}\right] \tag{4.5}$$

```
> IsOrthogonal(M);
```

$$\mathit{true} \tag{4.6}$$

Solve the system defined by Matrix **M** and Vector **v**.

```
> M:=Matrix([[1, 1, 3, -1], [1, 1, 1, 1], [1, -2, 1, -1], [4,
  1, 8, -1]]);
```

$$M := \left[\begin{array}{cccc} 1 & 1 & 3 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -2 & 1 & -1 \\ 4 & 1 & 8 & -1 \end{array}\right] \tag{4.7}$$

```
> v:=Vector([0, 1, 1, 0]);
```

$$v := \left[\begin{array}{c} 0 \\ 1 \\ 1 \\ 0 \end{array}\right] \tag{4.8}$$

```
> LinearSolve(M, v);
```

(4.9)

$$\begin{bmatrix} \frac{25}{6} \\ \frac{4}{3} \\ -\frac{5}{2} \\ -2 \end{bmatrix} \tag{4.9}$$

Construction of simple Matrices and Vectors, extraction of submatrices, transposition, basic arithmetic and computation of inner products can be done directly without use of commands in the LinearAlgebra package.

```
> u := Vector([1,3]);
```

$$u := \begin{bmatrix} 1 \\ 3 \end{bmatrix} \tag{4.10}$$

```
> v := Vector([5,7]);
```

$$v := \begin{bmatrix} 5 \\ 7 \end{bmatrix} \tag{4.11}$$

```
> u+v;
```

$$\begin{bmatrix} 6 \\ 10 \end{bmatrix} \tag{4.12}$$

```
> u.v;
```

$$26 \tag{4.13}$$

```
> A := Matrix([[1,3],[5,7]]);
```

$$A := \begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix} \tag{4.14}$$

```
> B := Matrix([[1,1],[1,1]]);
```

$$B := \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \tag{4.15}$$

```
> A+2*B;
```

$$\begin{bmatrix} 3 & 5 \\ 7 & 9 \end{bmatrix} \tag{4.16}$$

```
> A.B;
```

$$\begin{bmatrix} 4 & 4 \\ 12 & 12 \end{bmatrix} \tag{4.17}$$

```
> A.u;
```

$$\begin{bmatrix} 10 \\ 26 \end{bmatrix} \tag{4.18}$$

```
> A^(-1);
```

$$\begin{bmatrix} -\frac{7}{8} & \frac{3}{8} \\ \frac{5}{8} & -\frac{1}{8} \end{bmatrix} \tag{4.19}$$

```
> A^%T;
```

$$\begin{bmatrix} 1 & 5 \\ 3 & 7 \end{bmatrix} \tag{4.20}$$

Details

For more information including:

- a complete list of the routines in the **LinearAlgebra** package
- the supported data structures and data types
- the different sets of commands based on usage scenario: casual use or programming use
- the **LinearAlgebra[Modular]** subpackage for performing dense linear algebra computations in Z/m.
- the **LinearAlgebra[Generic]** subpackage for computing with generic implementations of algorithms for linear algebra over fields, Euclidean domains, integral domains and rings.

see the Details of the Linear Algebra Package help page.

See Also

LAIndex

examples/LA_Syntax_Shortcuts - LinearAlgebra package shortcuts.

VectorCalculus - a collection of commands that perform multivariate and vector calculus operations.